



Express Mail No. EU052134538US

#5/A  
5-13-02  
Dwyatt

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION OF : ROBERT M. MORRIS & LEET E. DENTON, III  
TITLE : A Visually Oriented Computer Implemented  
Application Development System Utilizing  
Standardized Objects And Multiple Views  
APPLICATION NO. : 09/184,738  
FILING DATE : November 2, 1998  
ART UNIT : 2122  
EXAMINER : Hoang-Vu Antony Nguyen-Ba  
ATTORNEY DOCKET NO. : 3042-3

RECEIVED  
MAY 13 2002  
Technology Center 2100

TO: Commissioner for Patents  
Washington, DC 20231

Dear Sir:

RESPONSE TO OFFICE ACTION DATED DECEMBER 6, 2001

Applicants hereby respond to the Office Action dated December 6, 2001. In the Office Action the Examiner has both objected to and rejected the claims. Applicants herewith present amended claims followed by remarks explaining the amendments and responding to the grounds for rejection.

Amendments to the claims are indicated by underlining for additions and bracketing for deletions. A clean copy of the amended claims and a separate marked up copy are attached to this Response. Please amend without prejudice the claims as follows:

1. A computer implemented system employing objects for generating an application script, in which both the objects and the script may be maintained separately, [without the necessity of specifying programatic steps in a character based representation] comprising:

a. means for wrapping [standardized] objects with additional properties and events beyond those properties and events internal to the [provided in the standardized] object; and

b. means for utilizing the additional properties and events to link and sequence the objects [into the application].

2. A computer implemented system employing objects for generating an application script, in which both the objects and the script may be maintained separately, representing [an application] a program structure [without the necessity of specifying programatic programmatic steps in a character based representation] comprising:

a. means for simultaneously displaying a plurality of different representations of the program structure; and

b. means for manipulating the program structure within each of the [four] different representations[;]

wherein the representations of the program structure may be synchronized [among the displays at the election of the user].

3. The system of claim 2 further comprising a means for highlighting the icon for each object [depiction of the objects] in the representations as [those] objects are being instantiated [realized] during application development playback preview.

4. A computer implemented system employing objects and utilizing a script, in which both the objects and the script may be maintained separately, [which does not require the necessity of specifying programatic steps in a character based representation] comprising:

a. a development environment and an interpreting run time [a playback] environment [that have no logical operators]; and

b. means for utilizing objects [,] by specifying property values according to the script[, standard objects].

5. The system of claim 4 further comprising a means for communicating among [standard] objects through the exchange of property values.

6. The system of claim 5 further comprising a means for communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

7. The system of claim 4 further comprising a means for communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

8. A computer implemented system employing objects and utilizing a script, in which both the objects and the script may be maintained separately, [which does not require the necessity of specifying programatic steps in a character based representation] comprising:

a. a development environment and an interpreting run time [a playback] environment that have no logical or arithmetic operators; and

b. means for utilizing objects [,] by specifying property values according to the script[, standard objects].

9. The system of claim 8 further comprising a means for communicating among [standard] objects through the exchange of property values.

10. The system of claim 9 further comprising a means for communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

11. The system of claim 8 further comprising a means for communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

12. A computer implemented system employing objects and utilizing a script, in which both the objects and the script may be maintained separately, [which does not require the necessity of specifying programatic steps in a character based representation] comprising:

a. a development environment and an interpreting run time [a playback] environment that have no definable data structure architecture; and

b. means for utilizing objects [,] by specifying property values according to the script [, standard] objects].

13. The system of claim 12 further comprising a means for communicating among [standard] objects through the exchange of property values.

14. The system of claim 13 further comprising a means for communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

15. The system of claim 12 further comprising a means for communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

16. The system of claim 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, or 15 further comprising a means for adding additional programming constructs by employing [standard] objects that perform the function [work] of programming constructs wherein unlimited expansion of program capabilities is achieved.

17. A computer implemented system employing objects and interpreting a script in which both the objects and the script may be maintained separately, comprising:

a. a run time program [that has no logical operators]; and

b. means for utilizing [standard] objects according to the script [by identifying the objects and specifying property values].

18. A computer implemented system employing objects and interpreting a script in which

both the objects and the script may be maintained separately, comprising:

- a. a run time program that has [no] neither arithmetic nor logical operators; and
- b. means for utilizing [standard] objects according to the script [by identifying the objects and specifying property values].

19. A computer implemented system employing objects and interpreting a script in which both the objects and the script may be maintained separately., comprising:

- a. a run time program that has no definable data structure architecture; and
- b. means for utilizing [standard] objects according to the script [by identifying the objects and specifying property values].

20. A computer implemented development and run time system employing objects which utilizes a script, in which both the objects and the script may be maintained separately, utilizing a minimum set of core functionalities comprising:

- a. means for instantiating objects;
- b. means for integrating objects;
- c. means for sequencing objects; and
- d. means for providing communication among objects[; and
- e. means for synchronizing views]

wherein the [displayed] functionalities performed by the system during execution are determined by the [choice of] objects used and the script. [manner of their implementation in the system.]

21. A computer implemented run time system employing objects utilizing a minimum set of core functionalities which interprets a script, in which both the objects and the script may be maintained separately, comprising:

- a. means for instantiating objects;
- b. means for integrating objects;
- c. means for sequencing objects; and
- d. means for providing communication among objects;

wherein the [displayed] functionalities performed by the system during execution are determined by the [choice of] objects used and the script. [manner of their implementation in the system.]

22. A computer implemented system for [arranging] employing objects , having property values and event connections, which can be set in time and turned on or off of a visually perceptible display device comprising:

- a. means for setting the values of properties and connecting events;
- b. means for recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and
- c. means for traversing the history one change at a time

wherein the property values and event connections may be edited from any point in the history.

23. A computer implemented run time system employing objects which [does not require the necessity of specifying programatic steps in a character based representation that] interprets a script containing property values and event settings, in which both the objects and the script may be maintained separately, and dynamically executes objects comprising:

- a. means for wrapping [standardized] objects with additional properties and events beyond those properties and events internal to the [provided in the standardized] objects;
- b. means for utilizing the additional properties and events to link and sequence the

objects; and

c. means for reading one or more sets of property values and event settings maintained separately from the run time system and the objects

wherein the execution of the objects is determined [governed] by the property values and event settings in the script [property values].

24. The system of claim 23 further comprising means for adding programming constructs [and] or sub-languages utilizing objects.

25. A computer implemented system which interprets a script containing property values and event settings, which may be maintained separately, [which does not require the necessity of specifying programatic steps in a character based representation] that distributes processing to objects, provides and manages data flow among objects, and manages the execution [scheduling] of objects comprising:

a. means for wrapping [standardized] objects with additional properties and events beyond those properties and events internal to the [provided in the standardized] object; and

b. means for utilizing the additional properties and events to link and sequence the objects [;

c. means for specifying property values; and

d. means for saving the property values to a separate file]

wherein the run time execution of the objects is determined by [the] property values and events.

26. A computer implemented system employing objects which implements parallel processing [without the necessity of specifying programatic steps in a character based representation] comprising:

a. means for wrapping [standardized] objects with additional properties and events beyond those properties and events provided native to the [in the standardized] object;

b. means for utilizing the additional properties and events to link and sequence the objects; and

c. means for specifying the temporal relationship among [standard] objects by placing the objects on one or more time lines

wherein execution of the objects occurs at least partially concurrently and during which property values may be exchanged among the objects and events may be initiated.

27. An object oriented programming [A] computer implemented system in which the function of programming constructs is achieved by dynamically executing [utilization of standard] objects comprising:

a. means for wrapping [standardized] objects with additional properties and events beyond those properties and events provided internal to the [in the standardized] object;

b. means for utilizing the additional properties and events to link and sequence the objects [into the application]; and

c. means for specifying a list of property values and event settings

wherein the execution of the objects is determined by [a] the list of property values and event settings.

28. A computer implemented software method employing objects for generating an application script, in which both the objects and the script may be maintained separately, [without the necessity of specifying programatic steps in a character based representation] comprising the steps of:

a. wrapping [standardized] objects with additional properties and events beyond those



properties and events internal to the [provided in the standardized] object; and

b. utilizing the additional properties and events to link and sequence the objects [into the application].

29. A computer implemented software method employing objects for generating an application script, in which both the objects and the script may be maintained separately, representing [an application] a program structure [without the necessity of specifying programmatic steps in a character based representation] comprising the steps of:

a. simultaneously displaying a plurality of different representations of the program structure; and

b. means for manipulating the program structure within each of the [four] different representations[;]

wherein the representations of the program structure may be synchronized [among displays at the election of the user].

30. The software method of claim 29 further comprising the step of highlighting the icon for each object [depiction of the objects] in the representations as [those] objects are being instantiated [realized] during application development run time [playback] preview.

31. A computer implemented software method employing objects and utilizing a script, in which both the objects and the script may be maintained separately, [for programming a computer which does not require the necessity of specifying programmatic steps in a character based representation] comprising the steps of:

a. utilizing a development environment and an interpreting run time [a playback] environment [that have no logical operators]; and

b. utilizing objects [,] by specifying property values according to the script[, standard

objects].

32. The software method of claim 31 further comprising the step of communicating among [standard] objects through the exchange of property values.

33. The software method of claim 32 further comprising the step of communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

34. The software method of claim 31 further comprising the step of communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

35. A computer implemented software method employing objects and utilizing a script, in which both the objects and the script may be maintained separately, [for programming a computer which does not require the necessity of specifying programatic steps in a character based representation] comprising the steps of:

a. utilizing a development environment and an interpreting run time [a playback] environment that have no logical or arithmetic operators; and

b. utilizing objects [,] by specifying property values according to the script[, standard objects].

36. The software method of claim 35 further comprising the step of communicating among [standard] objects through the exchange of property values.

37. The software method of claim 36 further comprising the step of communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

38. The software method of claim 35 further comprising the step of communicating among

[standard] objects wherein an event generated by an object triggers an instance of [an] another object.

39. A computer implemented software method employing objects and utilizing a script, in which both the objects and the script may be maintained separately, [which does not require the necessity of specifying programatic steps in a character based representation] comprising the steps of:

a. utilizing a development environment and an interpreting run time [a playback] environment that have no definable data structure architecture; and

b. utilizing objects [,] by specifying property values according to the script [, standard] objects].

40. The software method of claim 39 further comprising the step of communicating among [standard] objects through the exchange of property values.

41. The software method of claim 40 further comprising the step of communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

42. The software method of claim 39 further comprising the step of communicating among [standard] objects wherein an event generated by an object triggers an instance of [an] another object.

43. The software method of claim 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, or 42 further comprising the step of adding additional programming constructs by employing [standard] objects that perform the function [work] of programming constructs wherein unlimited expansion of program capabilities is achieved.

44. A computer implemented software method employing objects and interpreting a script in

which both the objects and the script may be maintained separately, for executing an application comprising the steps of:

- a. utilizing a run time program [that has no logical operators]; and
- b. utilizing [standard] objects according to the script [by identifying the objects and specifying property values].

45. A computer implemented software method employing objects and interpreting a script in which both the objects and the script may be maintained separately, for executing an application comprising the steps of:

- a. utilizing a run time program that has [no] neither arithmetic nor logical operators; and
- b. utilizing [standard] objects according to the script [by identifying the objects and specifying property values].

46. A computer implemented software method employing objects and interpreting a script in which both the objects and the script may be maintained separately, for executing an application comprising the steps of:

- a. utilizing a run time program that has no definable data structure architecture; and
- b. utilizing [standard] objects according to the script [by identifying the objects and specifying property values].

47. A computer implemented development and run time software method employing objects for developing and executing an application which utilizes a script, in which both the objects and the script may be maintained separately, and utilizing a minimum set of core functionalities comprising the steps of:

- a. instantiating objects;

- b. integrating objects;
- c. sequencing objects; and
- d. providing communication among objects[; and
- e. synchronizing views]

wherein the [displayed] functionalities performed by the software method during execution are determined by the [choice of] objects used and the script [manner of their implementation in the system].

48. A computer implemented run time software method employing objects for executing an application utilizing a minimum set of core functionalities which interprets a script, in which both the objects and the script may be maintained separately, comprising the steps of:

- a. instantiating objects;
- b. integrating objects;
- c. sequencing objects; and
- d. providing communication among objects;

wherein the [displayed] functionalities performed by the software method during execution are determined by the [choice of] objects used and the script. [manner of their implementation in the system.]

49. A computer implemented software method for [arranging] employing objects , having property values and event connections, which can be set in time and turned on or off of a visually perceptible display device comprising the steps of:

- a. setting the values of properties and connecting events;
- b. recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and

c. traversing the history one change at a time

wherein the property values and event connections may be edited from any point in the history.

50. A computer implemented run time software method employing objects which [does not require the necessity of specifying programatic steps in a character based representation that] interprets a script containing property values and event settings, in which both the objects and the script may be maintained separately, and dynamically executes the objects comprising the steps of:

a. wrapping [standardized] objects with additional properties and events beyond those properties and events native to the [provided in the standardized] objects;

b. utilizing the additional properties and events to link and sequence the objects; and

c. reading one or more sets of property values and event settings maintained separately from the run time system and the objects

wherein the execution of the objects is determined [governed] by the property values and event settings in the script [property values].

51. The software method of claim 50 further comprising the step of adding programming constructs [and] or sub-languages utilizing objects.

52. A computer implemented software method which interprets a script containing property values and event settings, which may be maintained separately, [which does not require the necessity of specifying programatic steps in a character based representation] that distributes processing to objects, provides and manages data flow among objects, and manages the execution [scheduling] of objects comprising the steps of:

a. wrapping [standardized] objects with additional properties and events beyond those

properties and events internal to the [provided in the standardized] object; and

- b. utilizing the additional properties and events to link and sequence the objects [;
- c. means for specifying property values; and
- d. saving the property values to a separate file]

wherein the [run time] execution of the objects is determined by the property values and events.

53. A computer implemented software method employing objects which implements parallel processing [without the necessity of specifying programatic steps in a character based representation] comprising the steps of:

- a. wrapping [standardized] objects with additional properties and events beyond those properties and events provided internal to the [in the standardized] object;
- b. utilizing the additional properties and events to link and sequence the objects; and
- c. specifying the temporal relationship among [standard] objects by placing the objects on one or more time lines

wherein execution of the objects occurs at least partially concurrently and during which property values may be exchanged among the objects and events may be initiated.

54. A computer implemented object oriented software programming method in which the function of programming constructs is achieved by dynamically executing [utilization of standard] objects comprising the steps of:

- a. wrapping [standardized] objects with additional properties and events beyond those properties and events provided internal to the [in the standardized] object;
- b. utilizing the additional properties and events to link and sequence the objects [into the application]; and